

Code 582

Flight Software Branch

FSB GUIDELINES FOR REQUIREMENTS MANAGEMENT IN MKS

Flight Software Branch – Code 582

Version 1.0 – 08/23/07

582-2007-023



Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

UPDATE HISTORY

Version	Date	Description	Affected Pages
1.0	8/23/07	Baseline – K. Narayanan	All

CONTENTS

<i>Introduction.....</i>	<i>1</i>
<i>1. Develop Requirements</i>	<i>1</i>
<i>Option 1 – Generate / Update FSW Requirements Document (WORD)</i>	<i>1</i>
<i>Option 2 – Generate / Update Requirements Directly in MKS</i>	<i>1</i>
<i>2. Freeze Build Requirements and Conduct Build Start Review (for each Build).....</i>	<i>2</i>
<i>3. Manage Links to Development Tasks and Tests (for each Build)</i>	<i>2</i>
<i>Development</i>	<i>2</i>
<i>Test</i>	<i>3</i>
<i>4. Manage Requirement Changes</i>	<i>3</i>
<i>Appendix – Workflow and State Transitions</i>	<i>5</i>

Introduction

This document provides guidelines and procedures for requirements management in the MKS environment. See [MKS Integrity Manager 2006 Users Guide](#) for additional information.

1. Develop Requirements

Option 1 – Generate / Update FSW Requirements Document (WORD)

- Generate a new (Master) document with draft requirements.
 - Use [ISD Requirements Development Process](#).
 - Use [FSB Requirements Document Template](#) for the document.
 - **Use internal requirement numbers – not MKS generated numbers – for all requirement references.** Do not modify the numbers of requirements from re-use library, if any.
 - Based on FSW Build Plan concentrate on finalizing requirements that are planned to be implemented in the next build; other requirements may remain in draft form. NOTE: Some requirements may be implemented in multiple builds.
 - Use the Master document for all subsequent new requirements. Document may be updated only for context and new requirements. **Once requirements are imported into MKS all updates to them have to be done directly in MKS.**
- Conduct Requirements Peer Reviews as needed and incorporate recommendations into the document.
 - Use [FSB Requirements Inspection Standard](#) to conduct the review.
 - For reference purpose, generate a report from MKS for those requirements already approved and in MKS, along with ancillary information such as **Planned Build** and **State**.
 - Review / update Build Assignments.
 - Ensure all requirements for next build are approved so they can be implemented.
- Import **new approved** requirements into MKS. These will be tagged by MKS and highlighted in the document. Requirements not approved in the review are not imported.
 - Assign **State** as 'Approved' for all new imported requirements. **NOTE: Any changes to approved requirements will require a Discrepancy Report / Change Request (DCR).**
 - Create Traceability links to parent and children requirements.
 - Assign **Planned Build**.

Option 2 – Generate / Update Requirements Directly in MKS

- Generate draft requirements in MKS.
 - Use [ISD Requirements Development Process](#).
 - Use [FSB Requirements Document Template](#) to develop Part 1 (up-front contextual part) of the requirements document.
 - Enter requirements – initial **State** is 'Proposed.' Requirements may be revised indefinitely in this state, and no DCR will be needed.
 - Assign unique requirement numbers. **Use internal requirement numbers – not MKS ID – for all requirement references. Do not modify the numbers of requirements from re-use library, if any.**
 - Concentrate on requirements that need to be implemented in the next build – other requirements may remain in draft form (**State** = 'Proposed'). Some requirements may be implemented in multiple builds.

- Create Traceability links to parent and children requirements.
 - Assign **Planned Build**.
- Conduct Requirements Peer Reviews as needed and incorporate recommendations into the requirements.
 - Generate a report from MKS for review. For reference include other previously approved requirements in MKS as well. Include ancillary information such as *Planned Build* and *State*.
 - Use *FSB Requirements Inspection Standard* to conduct the review.
 - Assign **State** as 'Approved' for all new requirements that are approved.

NOTE: Any changes to approved requirements will require DCR.

 - Review / update **Planned Build**.

2. Freeze Build Requirements and Conduct Build Start Review (for each Build)

- Assign **Actual Build** to new requirements in the Build.
- Generate a Build Requirements review document from MKS.
 - Include **State**, **Planned Build**, **Actual Build**, **Revision Build**.
 - Include only current Build. However, it may be useful to review requirements planned for future Builds as well, so that some requirements could be reassigned if necessary.
 - Include already implemented requirements that have changed and need to be re-implemented in the Build (**Revision Build** set to current Build, and **State** changed to 'Revised' and 'Approved').
 - Include DCRs on requirements that failed testing in previous build. More DCRs may be assigned to the Build during the Build development.
- Review the Build requirements with Build stakeholders – Test Team, Developers, and possibly others. Incorporate review comments into MKS.
- Generate Build Requirements Report from MKS – assign a version number and date, put it in CM.

NOTE

Ideally, Build development should start only after this review, but it may start earlier with the understanding that the delivered Build MUST implement the APPROVED requirements for that Build. Similarly Test Team may develop scenarios and test procedures using draft requirements but the final procedures MUST verify APPROVED requirements.

3. Manage Links to Development Tasks and Tests (for each Build)

Development

- Create an Implementation DCR for each new development task for the Build. Set up links between the DCR and the new requirements assigned to the Task for the Build.

NOTE: Revised requirements are linked to other implementation DCRs.
- Implement each Task. Link any new requirements that come in late, and assign **Actual Build**.
- After Build Integration, when Task Implementation DCR state changes to 'Ready for Test,' change **State** of all linked Requirements from 'Approved' to 'Implemented.'

TIP: This may be done for all the Build requirements (for all tasks) at once, since all tasks for the build are delivered at the same time.

- Change **State** of Requirements revised in the current Build to ‘Implemented.’
- Review all requirements for the Build. Add, remove, or update their **State**, **Actual Build**, **Revision Build**, and links as necessary.
- Create a snapshot report of the requirements as delivered in the Build and attach it to the VDD. Also save it as an attachment with the Requirements Documents issue in MKS.

Test

- Create new Test issues for the new Build. Set up links between Test issue and Build requirements.
NOTE: Typically several Tests are developed to test all requirements for a Task. Possibly multiple requirements are tested by each Test, and, some requirements may be tested by multiple Tests.
- Revise Tests for the previous Builds and their requirement links, as necessary.
- Before final Test execution all linked requirements should have the correct **Actual Build** – if not, update Test links and Tests.
- Execute Test. Each Test execution may pass some requirements and fail others. Set **Build Test Status** to ‘Pass,’ ‘Fail,’ ‘Partial Pass’ , or ‘Untestable at this time’ depending on Test result. Set **Tested Build**.
NOTE: Some requirements may be tested by multiple tests, and so **Test Status** should be the aggregate of all applicable tests.
- Submit Failed Test DCRs with links to Test and failed requirements.
- After Build Test is complete take a snapshot of RTTMs and archive them with date.

NOTE

The requirements in MKS are dynamic and may be revised anytime. The fields, **Planned Build**, **Actual Build**, **Revision Date**, and **Revision Build**, along with the requirement **State**, give a good indication of the requirements’ development status. (**Planned Build** and **Actual Build** are never changed once the requirement is delivered in a Build.)

For each Build the Test Team must use the **snapshot of the delivered Build requirements** from the VDD for their tests. **Tested Build** and **Build Test Status** reflect the status of the **requirements as delivered in the Tested Build**. The Historic Reports generation feature in MKS may be used to view requirement changes since the build delivery date.

4. Manage Requirement Changes

Requirement changes or deletions when **State** = ‘Propose’:

- No DCR is necessary; make changes or deletions as needed.
- Requirements and changes are not counted for metrics.

Requirement Change DCRs that result in FSW change:

- Coordinate through IRB and get approval – there may be impact to Development, or Test, or both.
- Set Requirement **State** to ‘Revise’ and add **Revision Comment/Revision Date/ Revision Build**.
- Make the change and link the DCR. NOTE: IRB approval is preferred.
- Set Requirement **State** to ‘Approved’.

- Generate related (linked) FSW DCR and proceed with its development.

Requirement Change DCRs that are purely editorial and do not result in FSW change:

- Discuss at IRB and get approval. **NOTE:** Trivial typographical changes may be done without a DCR and IRB approval.
- Note current Requirement *State*.
- Set *State* to 'Revise' and add *Revision Comment/Date/Revision Build*.
- Make the change, link the DCR, and restore original Requirement *State*.

Requirement Deletion DCRs:

- Discuss at IRB and get approval.
- Set Requirement *State* to 'Revise' and add *Revision Comment/Date/Revision Build*.
- Remove traceability links.
- Add '**DELETED:** ' before the requirement, link the DCR, and set Requirement *State* to 'Cancel.'

Appendix – Workflow and State Transitions

This appendix shows the requirement workflow and its relationship to the Task and Test workflows.

- Requirement states – ‘Propose,’ ‘Approved,’ ‘Implemented,’ ‘Revise,’ and ‘Cancel.’ See Figure 1 for state transitions and workflow dependencies among Requirements, Tasks, and Tests.
- When a Requirement state changes from any state to ‘Revise’ all related children requirements and TESTs become suspect.

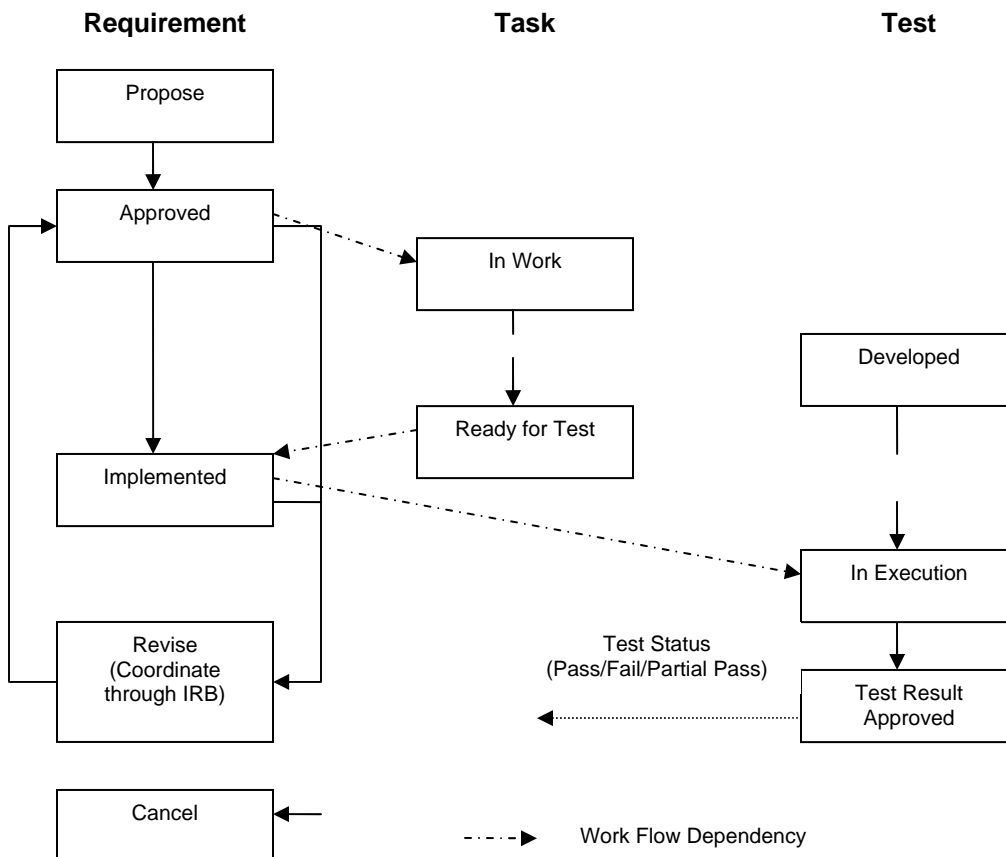


Figure 1. Work Flows and State Transitions